

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-337736

(43)Date of publication of application : 07.12.2001

(51)Int.Cl. G06F 1/04

G06F 1/32

(21)Application number : 2000- (71)Applicant : SEIKO EPSON CORP
158010

(22)Date of filing : 29.05.2000 (72)Inventor : WAKABAYASHI SHUICHI

(54) COMPUTER SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a computer system capable of reducing consumption current without deterioration of performance.

SOLUTION: The computer system includes a CPU clock generator for the CPU, bridge device, RAM, ROM, input device, output device, communication processing device, external storage device, external bus, clock controlling bus, internal bus and clock controlling device that controls clock signals starting/ stopping the CPU.

The clock controlling device includes a bridge interface device 20, control device 21, counter A 22, resistor A 23, comparator A 30, counter B 24, resistor B 25, comparator B 31, resistor C 26, resistor D 27, event monitoring device 28, resistor E 29, STOPCLK signal generator 32, CPU controlling device 33, AND gate 34 and NOR gate 35.

CLAIMS

[Claim(s)]

[Claim 1] A computer system wherein said clock control means is provided with a clock setting-out means to set up said operation/stop clock signal arbitrarily according to load of said CPU in a computer system characterized by comprising the following.

It is controllable CPU (Central Processing Unit) with a signal from the outside about its own operation/stop.

A means to input operation/stop of said CPU as an operation/stop clock signal from the exterior.

A clock control means to control said operation/stop clock signal.

[Claim 2] The computer system according to claim 1 wherein said clock setting-out means sets up said operation/stop clock based on a division ratio or a duty ratio of a reference clock.

[Claim 3] The computer system according to claim 1 to 2 wherein said clock setting-out means sets the aforementioned from operation / stop clock as a fixed value of ON or OFF by a ratio according to load of said CPU.

[Claim 4] The computer system according to claim 1 to 3 when said clock setting-out means occurs [event] such as interruption wherein it has a clock revolution means which uses as a reference clock said operation/stop clock set up by said clock setting-out means.

[Claim 5] The computer system according to claim 4 wherein said clock setting-out means generates interruption to predetermined timing above.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention for example about the computer system of a personal computer a personal digital assistant etc. It is related with the computer system which can control the operation clock (only henceforth a "CPU clock") of CPU especially supplied to CPU (Central Processing Unit).

[0002]

[Description of the Prior Art] In recent years portable computer systems such as a portable personal computer and a personal digital assistant are developed variously. Since these portable computer systems are driven with a battery in many cases low power consumption is desired. However CPU carried in these computer systems is accelerated every year and the power consumption of CPU is increasing steadily.

[0003] The conventional computer system is provided with the clock control part for reducing the consumed electric current of CPU else [such as CPU ROM (Read Only Memory) RAM (Random Access Memory) and an input/output device]. In such a conventional computer system When the load of CPU is large a clock control part A CPU clock stop signal (STOPCLK#) is turned OFF CPU is operated based on a high-speed operation clock and when the load of CPU is small a CPU clock stop signal (STOPCLK#) is turned ON and operation of CPU is stopped temporarily. Thus in the conventional computer system when a clock control part controls supply of a CPU clock based on a CPU clock stop signal (STOPCLK#) and operates CPU efficiently the consumed electric current of CPU is reduced.

[0004]

[Problem(s) to be Solved by the Invention] However in the conventional computer system the clock control part turned on and off the CPU clock stop signal (STOPCLK#) and was controlling supply of the CPU clock. Therefore if the load of CPU becomes small in the case of the communications processing etc. which perform serial processing of data a clock control part turns ON a CPU clock stop signal (STOPCLK#) stopping operation of CPU temporarily and data taking and spilling -- etc. -- the error occurred and there was a problem that the performance of a computer system will fall by this error handling.

[0005]In order not to reduce the performance of the whole computer system as mentioned above when a CPU clock stop signal (STOPCLK#) was always turned off at the time of the serial processing of data there was a problem that the consumed electric current could hardly decrease.

[0006]Then it is providing the computer system which can reduce the consumed electric current without the purpose of this invention reducing performance in view of the above-mentioned point.

[0007]

[Means for Solving the Problem]In order to solve the above technical problem a computer system of this inventionIn a computer system provided with a clock control means to control operation/stop clock signal of controllable CPU (Central Processing Unit) and CPU for its own operation/stop by a signal from the exteriorA clock control means is characterized by having a clock setting-out means to set up operation/stop clock signal arbitrarily according to load of CPU.

[0008]In an above-mentioned computer system the clock setting-out means can also set up operation/stop clock signal based on a division ratio or a duty ratio of a reference clock.

[0009]In an above-mentioned computer system the clock setting-out means can also set operation/stop clock signal as ON or OFF by a ratio according to load of CPU.

[0010]Since operation/stop clock signal can be arbitrarily set up according to load of CPU CPU can be operated efficiently.

[0011]In an above-mentioned computer system the clock control means can also have a clock revolution means which uses as a reference clock operation/stop clock signal set up by a clock setting-out means when events such as interruption occur.

[0012]In an above-mentioned computer system the clock control means can also generate interruption to predetermined timing for a clock revolution means.

[0013]Since operation/stop clock signal can be immediately changed into a reference clock at the time of event generation such as interruption performance

is not reduced by error handling etc.

[0014]

[Embodiment of the Invention] Hereafter it explains referring to the drawing about the computer system of this invention.

[0015] Drawing 1 is a figure showing an example of the composition of the computer system of this invention. In drawing 1 the computer system of this invention CPU (Central Processing Unit) 1 which controls the whole system The CPU operation clock generating part 3 which supplies an operation clock to CPU 1 The clock control part 2 which controls operation/stop clock signal of CPU 1 RAM (Random Access Memory) 5 which memorizes data etc. ROM (Read Only Memory) 6 which has memorized OS (Operating System) etc. The input parts 7 such as a keyboard the outputting parts 8 such as a liquid crystal display and the communication processing part 9 that connect with an external network etc. and transmits and receives data The bridge part 4 which separates the external storages (HDD etc.) 10 access to the I/O register of the clock control part 2 and access to RAM 5 and access to each formation parts 6-10 and generates each access timing It has the internal bus 13 which connects CPU 1 and the bridge part 4 and the clock control part 2 the external bus 11 which connects the bridge part 4 and each formation parts 6-10 and the clock control bus 12 which exchanges data between the bridge part 4 and the clock control part 2.

[0016] In this computer system CPU 1 executes the program memorized by ROM 6 and the external storage 10 based on the signal into which it is inputted from the input part 7 the communication processing part 9 etc. At this time the clock control part 2 supervises the load of CPU 1 and controls operation/stop signal of CPU 1 by software or hardware according to that situation.

[0017] Drawing 2 is a block diagram showing the composition of the clock control part 2 of this invention. The bridge I/F (Inter/Face) part 20 from which this clock control part 2 serves as a command with CPU 1 (drawing 1) in bridge part 4 (refer to drawing 1) course and I/F (Inter/Face) of delivery of data in drawing 2 The control section 21 which controls the clock control part 2 whole and the CPU

control part 33 which controls operation/stop of CPU1 (drawing 1)The counter A22 which calculates a reference clock in order to measure operation timingThe register A23 which holds a non-operating ratio value among the intermittent-control-action ratios of CPU1 (drawing 1) set up by the control section 21The comparator A30 which compares the preset value of not operating of the register A23 with the counted value outputted from the counter A22The counter B24 which calculates a reference clock in order to measure operation timingThe register B25 which holds a ratio value of operation among the intermittent-control-action ratios of CPU1 (drawing 1) set up by the control section 21The comparator B31 which compares the preset value of operation of the register B25 with the counted value outputted from the counter B24and the register C26 which outputs an intermittent-control-action demand (SLOWCLKREQ) signal based on the control signal from the control section 21The register D27 which outputs an operation stop demand (STOPCLKREQ) signal based on the control signal from the control section 21The interrupt signal and the input part 7 (drawing 1) from the outsidethe outputting part 8 (drawing 1)the communication processing part 9 (drawing 1)The event Monitoring Department 28 which outputs to the register D27 by making the event signal (EVENT#) according to event informationsuch as I/O access to the external storage 10 (drawing 1) etc.into a trigger signalThe register E29 which leads and holds event information from the event Monitoring Department 28 based on the control signal of the control section 21The clock stop signal (STOPCLK#) for stopping operation of CPU1 (drawing 1) based on the signal outputted from the comparator A30the comparator B31the register C26and the register D27 is generated. The STOPCLK# signal generating part 32 outputted to the CPU control part 33 etc.AND gate 34 which carries out the logical product of the inversion signal of the GRANT signal from the CPU control part 33and the inversion signal of the STOPCLK# signal from the STOPCLK# signal generating part 32It has NOR gate 35 which carries out the logical sum of the output signal from AND gate 34and the STOPCLK# signal from the STOPCLK# signal generating part 32and

carries out an inverted output.

[0018]Herebased on the division ratio and duty ratio of a reference clock at the time of CPU1 (drawing 1) operationthe ratio of the intermittent control action held at the register A23 and the register B25 can be pushed in the bridge I/F (Inter Face) part 20and software etc. can determine it programmably.

[0019]The CPU control part 33 outputs CPUSTOPCLK# to CPU1 (drawing 1) based on the STOPCLK# signal outputted from the STOPCLK# signal generating part 32. The CPU control part 33 outputs a ready signal (RDY#) to CPU1 (drawing 1) while it receives the state of CPU1 (drawing 1) from CPU1 (drawing 1) through the internal bus 13 and outputs a GRANT signal to AND gate 34.

[0020]The STOPCLK# signal with which the counter A22 is outputted to the CPU control part 33 from the STOPCLK# signal generating part 32 is active ("low" level)And only when the GRANT signal outputted from the CPU control part 33 is active ("high" level) (that iswhile a STOPCLK# signal is a "low" level and CPU1 has stopped)it continues a count. When the output signal from the STOPCLK# signal generating part 32 and the CPU control part 33 is except the above,terminal CLR serves as a "low" level and the counter A22 resets a count. On the other handthe counter B24 continues a countwhen the STOPCLK# signal outputted to the CPU control part 33 from the STOPCLK# signal generating part 32 is inactive ("high" level)A count is reset when a STOPCLK# signal is active ("low" level).

[0021]If the comparator A30 compares with the preset value of the register A23 the counted value which is an output value from the counter A22 and counted value and its preset value correspondit will output a "high" level to the generator terminal of the STOPCLK# signal generating part 32. On the other handif the comparator B31 compares with the preset value of the register B25 the counted value which is an output value from the counter B24 and counted value and its preset value correspondit will output a "high" level to the battery terminal of the STOPCLK# signal generating part 32.

[0022]the EVENTCLR# signal with which the event Monitoring Department 28 is

outputted from the register E29 -- being active ("low" level) -- it is reset when it becomes. the EVENT# signal with which the register D27 is outputted from the event Monitoring Department 28 -- being active ("low" level) -- it will be reset if it becomes.

[0023]Drawing 3 is a figure showing the concrete circuitry of the STOPCLK# signal generating part 32. The STOPCLK# signal generating part 32 (drawing 2)The toggle circuit 40 which outputs a signal based on the pulse signal from the comparator A30 and the comparator B31 (drawing 2)AND gate 41 which inputs and carries out the logical product of the SLOWCLKREQ signal outputted from the output signal and the register C26 (drawing 2) from the toggle circuit 40It comprises NOR gate 42 which inputs and carries out the logical product of the STOPCLKREQ signal outputted from the signal outputted from AND gate 41and the register D27 (drawing 2)and carries out an inverted output.

[0024]The toggle circuit 40 fixes an output to a "high"if a "high" pulse signal is outputted from the comparator A30 (drawing 2)and if a "high" pulse signal is outputted from the comparator B31 (drawing 2)it fixes an output to a "low."

[0025]Hereafterthe function of the computer system shown by drawing 1 - drawing 3 is explained.

[0026]Drawing 4 is a key map showing the relation of the operating state of CPU1 and operational mode in the computer system of this invention. The operating state of CPU1 is classified into a "busy state" and an "idle state" as shown in drawing 4. A "busy state" is when the load of CPU1 is large like image processingor character recognition and numerical computationand an "idle state" is when loads of CPU1such as a keystroke waiting stateare small.

[0027]The control section 21 detects the operating state of CPU1 given from the bridge part 4 (Step 401)and judges whether the operating state of CPU1 changed (Step 402). Processing is ended if the operating state of CPU1 is not changing. When the operating state of CPU1 is changingthe operating state of present CPU1 is judged (Step 403). When the operating state of CPU1 is a "busy state"processing of the (A) busy mode is performed (Step 404). In the case of an

"idle state"an "idle state" is judged based on predetermined thresholds (operating ratio of CPUetc.) (Step 405)and it performs processing in (B) drop dead halt mode (Step 406) or (C) intermittent-control-action mode (Step 407)respectively. [0028]The control section 21 judges the operational mode of CPU1and an idol's condition from the operating ratio etc. which are the operating states of detected CPU1 based on a predetermined threshold. For exampleit may be made to judge the state of CPU1 by comparing the threshold and operating ratio which were beforehand determined based on the system configuration by making the division ratio and duty ratio of a reference clock into an operating ratio.

[0029]Belowthe flow of processing of the computer system of this invention in each CPU operational mode is explained in detail.

[0030](A) Busy mode drawing 5 is a flow chart which shows the flow of processing of the computer system of this invention in a busy mode. the SLOWCLKREQ signal which the control section 21 outputs a control signal to the register C26and is outputted from the register C26 in drawing 5 -- being inactive ("low" level) -- it resets (Step 501). ThereforeCPU1 works not by an intermittent control action or stop operation but by a normal operating state.

[0031](B) Drop dead halt mode drawing 6 is a flow chart which shows the flow of processing of the computer system of this invention in drop dead halt mode. the EVENTCLR# signal which the control section 21 outputs a control signal to the register E29and is first outputted from the register E29 in drawing 6 -- being active ("low" level) -- it is made to change and the event Monitoring Department 28 is reset (Step 601).

[0032]Nextif the control section 21 outputs a control signal to the register E29the register E29 will output an EVENTCLR# signal to the event Monitoring Department 28 in the inactive ("high" level) state. At this timethe reset state of the event Monitoring Department 28 is canceled (Step 602).

[0033]nextthe case (Step 603) where an event is detected at the event Monitoring Department 28 -- the event Monitoring Department 28 -- an EVENT# signal -- being active ("low" level) -- it setsthe register D27 is made into a reset

state (Step 604) and it does not go into drop dead halt mode but processing is ended.

[0034] the STOPCLKREQ signal outputted from the register D27 with the control signal from the control section 21 on the other hand when the event is not detected at the event Monitoring Department 28 (Step 603) -- being active ("high" level) -- it becomes (Step 605). If the STOPCLK# signal generating part 32 receives the "high" level of the STOPCLKREQ signal outputted from the register D27 by NOR gate 42 -- STOPCLK# -- being active ("low" level) -- it becomes and the signal is outputted to the CPU control part 33 (Step 606). The CPU control part 33 will perform predetermined CPU stop processing if STOPCLK# of an active ("low" level) state is received from the STOPCLK# signal generating part 32 (Step 607). Thereby the drop dead halt of CPU1 is carried out (Step 608).

[0035] if the event Monitoring Department 28 detects here an interrupt signal/O access etc. which were beforehand registered into event Monitoring Department 28 inside and which should be supervised (Step 609) -- the event Monitoring Department 28 -- an EVENT# signal -- being active ("low" level) -- it carries out (Step 610). the STOPCLKREQ signal which the register D27 will be reset if the signal of a "low" level is inputted into terminal CLR and is outputted from the register D27 -- being inactive ("low" level) -- it is reset (Step 611). receiving the event Monitoring Department 28 after saving event information inside on the other hand if the register E29 receives event information -- an EVENTCLR# signal -- being active ("low" level) -- it carries out and an EVENT# signal and event status are reset (Step 612). moreover -- if the event status signal of the event Monitoring Department 28 is reset as for the register E29 -- an EVENTCLR# signal -- being inactive ("high" level) -- it returns (Step 613). If the STOPCLKREQ signal ("low" level) outputted from the register D27 by NOR gate 42 is received the STOPCLK# signal generating part 32 According to the signal of the "low" level from AND gate 41 STOPCLK# will be in an inactive ("high" level) state and will be outputted to the CPU control part 33 (Step 614). The CPU control part 33 will perform CPU operation start processing if STOPCLK# of an

inactive ("high" level) state is received from the STOPCLK# signal generating part 32 (Step 615). Thereby CPU1 starts operation (Step 616).

[0036](C) Intermittent-control-action mode drawing 7 is a flow chart which shows the flow of processing of the computer system of this invention in intermittent-control-action mode. In drawing 7 first the control section 21 sets up an intermittent-control-action ratio based on the operating state of CPU1 given from the bridge part 4 and outputs a non-operating ratio value and the ratio value of the operation to the register B25 to the register A23 (Step 701). Based on the value of the register A23 and the register B25 the comparator A30 and the comparator B31 operate and each comparison result signal is outputted to the toggle circuit 40 of the STOPCLK# signal generating part 32. Here a comparison result signal serves as a "high" level when the preset value of a register and the value of a counter are in agreement. The toggle circuit 40 outputs the signal of a "high" level by the signal from the comparator A30 and outputs the signal of a "low" level by the signal from the comparator B31.

[0037] next the SLOWCLKREQ signal which the control section 21 outputs a control signal to the register C26 and is outputted from the register C26 -- being active ("high" level) -- it sets up (Step 702). In the STOPCLK# signal generating part 32 AND gate 41 outputs the signal of a "high" level according to the output signal of the "high" level from the toggle circuit 40 when the SLOWCLKREQ signal outputted from the register C26 is active ("high" level). NOR gate 42 outputs a STOPCLK# signal to the CPU control part 33 according to the STOPCLKREQ signal from the register D27 (Step 703).

[0038] the CPU control part 33 -- the STOPCLK# signal generating part 32 to a STOPCLK# signal -- being active ("low" level) -- if outputted (Step 704) predetermined CPU stop processing (Step 705) will be performed and CPU1 will suspend operation (Step 706).

[0039] When the interrupt signal from the outside I/O access etc. are detected at the event Monitoring Department 28 (Step 707) here the control section 21 the SLOWCLKREQ signal which outputs a control signal to the register 26 and is

outputted from the register 26 -- being inactive ("low" level) -- it resets (Step 713). If a SLOWCLKREQ signal is reset a STOPCLK# signal [being inactive ("high" level)] will be inputted into the CPU control part 33 and the CPU control part 33 will perform predetermined CPU operation start processing in it (Step 714). Thereby CPU1 starts operation (Step 715). The control section 21 cancels reset of the event Monitoring Department 28 after resetting the event Monitoring Department 28 (Step 716) (Step 717). This ends intermittent-control-action mode.

[0040] On the other hand in the state where neither the interrupt signal from the outside nor I/O access is detected at the event Monitoring Department 28 (Step 707) STOPCLK# inputted into the CPU control part 33 -- being inactive ("high" level) -- if it changes (Step 708) the CPU control part 33 will perform predetermined CPU operation start processing (Step 709). Thereby CPU1 starts operation (Step 710). At the event Monitoring Department 28 neither the interrupt signal from the outside nor I/O access is detected here (Step 711) a STOPCLK# signal -- being active ("low" level) -- when it becomes (Step 712) again by the CPU control part 33 predetermined CPU stop processing (Step 705) is performed and CPU1 suspends operation (Step 706). When the interrupt signal from the outside/I/O access etc. are detected at the event Monitoring Department 28 (Step 711) the control section 21 the SLOWCLKREQ signal which outputs a control signal to the register 26 and is outputted from the register 26 -- being inactive ("low" level) -- it resets (Step 713). If a SLOWCLKREQ signal is reset a STOPCLK# signal [being inactive ("high" level)] will be inputted into the CPU control part 33 and the CPU control part 33 will perform predetermined CPU operation start processing in it (Step 714). Thereby CPU1 starts operation (Step 715). The control section 21 cancels reset of the event Monitoring Department 28 after resetting the event Monitoring Department 28 (Step 716) (Step 717). This ends intermittent-control-action mode.

[0041] By the case where neither the interrupt signal from the outside nor I/O access is detected in Step 707 at the event Monitoring Department 28. When STOPCLK# is active ("low" level) (Step 708) the surveillance of an event is

successfully performed by the event Monitoring Department 28 (Step 707).

[0042] Similarly in Step 711 by the case where neither the interrupt signal from the outside nor I/O access is detected at the event Monitoring Department 28. When STOPCLK# is active ("low" level) (Step 715) the surveillance of an event is successfully performed by the event Monitoring Department 28 (Step 711).

[0043] Drawing 8 is a flow chart which shows the flow of predetermined CPU stop processing.

[0044] the STOPCLK# signal with which the CPU control part 33 is outputted from the STOPCLK# signal generating part 32 -- being active ("low" level) -- if it becomes it will output to CPU1 by making a CPUSTOPCLK# signal into an active state ("low" level) (Step 801). CPU1 -- a CPUSTOPCLK# signal -- being active ("low" level) -- if it becomes and a PAUSE command is received CPU status (STOPCLK# receipt state) will be outputted to the CPU control part 33 (Step 802). next the CPU control part 33 -- an RDY# signal -- being active ("low" level) -- it carries out and returns to CPU1 (Step 803). after 1CPUCLK and the CPU control part 33 -- an RDY# signal -- being inactive ("high" level) -- resetting -- a GRANT signal -- being active ("high" level) -- it sets (Step 804). Thereby CPU1 will be in a drop dead halt state.

[0045] Drawing 9 is a flow chart which shows the flow of predetermined CPU operation start processing.

[0046] the STOPCLK# signal with which the CPU control part 33 is outputted from the STOPCLK# signal generating part 32 -- being inactive ("high" level) -- if it becomes it will output to CPU1 by making a CPUSTOPCLK# signal into an inactive state ("high" level) (Step 901). the CPU control part 33 -- a GRANT signal -- being inactive ("low" level) -- it resets (Step 902). Thereby CPU1 starts operation.

[0047] Thus when the control section 21 judges the operating state of CPU according to the load of CPU1 and changes a CPU operation stop signal (STOPCLK#) in two or more modes CPU1 is linearly (gradual) controllable by from a drop dead halt state before a perfect operating state.

[0048]As mentioned abovealthough the computer system of this invention was explained(B) Interruption is generated from the control section 21 to predetermined timing to the event supervisory circuit 28and it may be made to make it return to the (A) busy mode from (B) drop dead halt mode or (C) intermittent-control-action mode at the time of drop dead halt mode or (C) intermittent-control-action mode.

[0049]Equip the control section 21 with a thermo sensora battery sensoretc.and the temperature and battery residual quantity of CPU1 are supervisedWhen exceeding a threshold with the signal from these sensorsit is made to shift to (C) intermittent-control-action mode or (B) drop dead halt mode compulsorilyand when falling from a threshold with a signalit may be made to make it return to the (A) busy mode.

[0050]

[Effect of the Invention]As mentioned aboveaccording to the computer system of this inventionthe consumed electric current could be reducedwithout reducing the performance of the computer system concerned.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a figure showing an example of the composition of the computer system of this invention.

[Drawing 2]It is a block diagram showing the composition of the clock control part of this invention.

[Drawing 3]It is a figure showing the circuitry of a STOPCLK# signal generating part.

[Drawing 4]It is a key map showing the relation of the operating state of CPU and operational mode in the computer system of this invention.

[Drawing 5]It is a flow chart which shows the flow of processing of a busy mode.

[Drawing 6] It is a flow chart which shows the flow of processing in drop dead halt mode.

[Drawing 7] It is a flow chart which shows the flow of processing of intermittent mode.

[Drawing 8] It is a flow chart which shows the flow of CPU stop processing.

[Drawing 9] It is a flow chart which shows the flow of CPU operation start processing.

[Description of Notations]

- 1 CPU
- 2 Clock control part
- 3 CPU operation clock generating part
- 4 Bridge part
- 5 RAM
- 6 ROM
- 7 Input part
- 8 Outputting part
- 9 Communication processing part
- 10 External storage
- 11 External bus
- 12 Clock control bus
- 13 Internal bus
- 20 Bridge I/F part
- 21 Control section
- 22 Counter A
- 23 Register A
- 24 Counter B
- 25 Register B
- 26 Register C
- 27 Register D
- 28 Event Monitoring Department

29 Register E
30 Comparator A
31 Comparator B
32 STOPCLK# signal generating part
33 CPU control part
3441 AND gates
3542 NOR gates
40 Toggle circuit

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-337736

(P2001-337736A)

(43)公産目 平成13年12月7日(2001.12.7)

(51)Int.Cl. ⁷	識別記号	FI	ページト*(参考)
G 0 6 F 1/04	3 0 1	G 0 6 F 1/04	3 0 1 C 5 B 0 1 1
1/32		1/00	3 3 2 Z 5 B 0 7 9

審査請求 有 請求項の数5 O.L (全11頁)

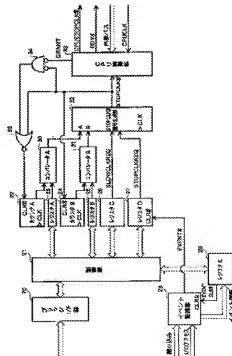
(21) 出願番号	特願2000-158010(P2000-158010)	(71) 出願人	000002369 セイコーエプソン株式会社 東京都新宿区西新宿2丁目4番1号
(22) 出願日	平成12年5月29日(2000.5.29)	(72) 発明者	若林 修一 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
		(74) 代理人	100095725 弁理士 上橋 雅彦 (外1名) Fターム(参考) 5B011 EA02 KK03 LL13 5B079 BA11 BB04 BC01 BC07

(54)【発明の名称】 コンピュータシステム

(57) 【要約】

【課題】 パフォーマンスを低下させることなく、消費電流を低減することができるコンピュータシステムを提供する。

【解決手段】 本発明のコンピュータシステムは、CPUと、CPU動作クロック発生部と、ブリッジ部と、RAMと、ROMと、入力部と、出力部と、通信処理部と、外部記憶装置と、外部バスと、クロック制御バスと、内部バスと、CPUの動作/停止クロック信号を制御するクロック制御部とを備え、クロック制御部は、ブリッジ/IF (Inter/IF) 部20と、制御部21と、カウンタA22と、レジスタA23と、コンパレータA30と、カウンタB24と、レジスタB25と、コンパレータB31と、レジスタC26と、レジスタD27と、イベント監視部28と、レジスタE29と、STOPCLK#信号生成部32と、CPU制御部33と、ANDゲート34と、NORゲート35とを備えている。



【特許請求の範囲】

【請求項1】 自分自身の動作/停止を外部からの信号で制御可能なCPU（Central Processing Unit）と、前記CPUの動作/停止を外部から動作/停止クロック信号として入力する手段と、前記動作/停止クロック信号を制御するクロック制御手段を備えたコンピュータシステムにおいて、前記クロック制御手段は、前記CPUの負荷に応じて前記動作/停止クロック信号を任意に設定するクロック設定手段を備えることを特徴とするコンピュータシステム。

【請求項2】 前記クロック設定手段は、基準クロックの分周比またはデューティ比に基づいて前記動作/停止クロックを設定することを特徴とする請求項1記載のコンピュータシステム。

【請求項3】 前記クロック設定手段は、前記CPUの負荷に応じた比率で前記動作/停止クロックをONまたはOFFの固定値に設定することを特徴とする請求項1乃至2記載のコンピュータシステム。

【請求項4】 前記クロック設定手段は、割り込みなどのイベントが発生した場合、前記クロック設定手段で設定した前記動作/停止クロックを基準クロックにするクロック回復手段を有することを特徴とする請求項1乃至3記載のコンピュータシステム。

【請求項5】 前記クロック設定手段は、前記に所定のタイムングで割り込みを発生することを特徴とする請求項4記載のコンピュータシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えばパーソナルコンピュータ、携帯型情報端末等のコンピュータシステムに関し、特に、CPU（Central Processing Unit）に供給されるCPUの動作クロック（以下、単に「CPUクロック」ともいう）を制御することができるコンピュータシステムに関する。

【0002】

【従来の技術】近年、携帯型パーソナルコンピュータや携帯型情報端末などの携帯型のコンピュータシステムが種々開発されている。これらの携帯型のコンピュータシステムは、バッテリーによって駆動することが多いため、低消費電力化が望まれている。ところが、これらコンピュータシステムに搭載されるCPUは年々高速化され、CPUの消費電力は増加する一方である。

【0003】従来のコンピュータシステムは、CPU、ROM（Read Only Memory）、RAM（Random Access Memory）、入出力装置などの他に、CPUの消費電流を低減させるためのクロック制御部を備えている。この様な従来のコンピュータシステムにおいては、CPUの負荷が大きき場合、クロック制御部は、CPUクロック停止信号（STOPCLK#）をOFFにしてCPUを高

速の動作クロックに基づいて動作させ、また、CPUの負荷が小さい場合には、CPUクロック停止信号（STOPCLK#）をONにしてCPUの稼働を一時的に停止させている。この様に、従来のコンピュータシステムにおいては、クロック制御部がCPUクロック停止信号（STOPCLK#）に基づいてCPUクロックの供給を制御し、CPUを効率良く動作させることによって、CPUの消費電流を低減させている。

【0004】

【発明が解決しようとする課題】しかしながら、従来のコンピュータシステムにおいては、クロック制御部がCPUクロック停止信号（STOPCLK#）のON/OFFを行ってCPUクロックの供給を制御していた。そのため、データのシリアル処理を行う通信処理などの場合、CPUの負荷が小さくなると、クロック制御部がCPUクロック停止信号（STOPCLK#）をONにして、CPUの稼働を一時的に停止してしまい、データの取りこぼし等のエラーが発生し、また、このエラー処理などによってコンピュータシステムのパフォーマンスが低下してしまうという問題があった。

【0005】また、上記の様にコンピュータシステム全体のパフォーマンスを低下させないために、データのシリアル処理時に常にCPUクロック停止信号（STOPCLK#）をOFFした場合、消費電流がほとんど低減できないという問題があった。

【0006】そこで、上記の点に鑑み、本発明の目的は、パフォーマンスを低下させることなく、消費電流を低減することができるコンピュータシステムを提供することである。

【0007】

【課題を解決するための手段】以上の課題を解決するため、本発明のコンピュータシステムは、自分自身の動作/停止を外部から信号で制御可能なCPU（Central Processing Unit）と、CPUの動作/停止クロック信号を制御するクロック制御手段を備えたコンピュータシステムにおいて、クロック制御手段は、CPUの負荷に応じて動作/停止クロック信号を任意に設定するクロック設定手段を備える事特徴とする。

【0008】また、上述のコンピュータシステムにおいて、クロック設定手段は、基準クロックの分周比またはデューティ比に基づいて動作/停止クロック信号を設定することもできる。

【0009】さらに、上述のコンピュータシステムにおいて、クロック設定手段は、CPUの負荷に応じた比率で動作/停止クロック信号をON又はOFFに設定することもできる。

【0010】CPUの負荷に応じて動作/停止クロック信号を任意に設定することができるので、CPUを効率よく動作させることができる。

【0011】また、上述のコンピュータシステムにおい

て、クロック制御手段は、割り込みなどのイベントが発生した場合、クロック設定手段で設定した動作/停止クロック信号を基準クロックにするクロック回帰手段を有することもできる。

【0012】さらに、上述のコンピュータシステムにおいて、クロック制御手段は、クロック回帰手段に所定のタイミングで割り込みを発生することもできる。

【0013】割り込みなどのイベント発生時に、動作/停止クロック信号を基準クロックに即座に変更することができるので、エラー処理などでパフォーマンスを低下させることがない。

【0014】

【発明の実施の形態】以下、本発明のコンピュータシステムについての図面を参照しつつ説明する。

【0015】図1は、本発明のコンピュータシステムの構成の一例を示す図である。図1において、本発明のコンピュータシステムは、システム全体を制御するCPU (Central Processing Unit) 1と、CPU1に動作クロックを供給するCPU動作クロック発生部3と、CPU1の動作/停止クロック信号を制御するクロック制御部2と、データなどを記憶するRAM (Random Access Memory) 5と、OS (Operating System) などを記憶しているROM (Read Only Memory) 6と、キーボードなどの入力部7と、液晶画面などの出力部8と、外部ネットワークなどに接続してデータの送受信を行う通信処理部9と、外部記憶装置 (HDDなど) 10と、クロック制御部2のI/OレジスタへのアクセスやRAM5へのアクセスと各構成部6~10へのアクセスを分離し、個々のアクセスタイミングを生成するブリッジ4と、CPU1と、ブリッジ4及びクロック制御部2とを接続する内部バス13と、ブリッジ4と各構成部6~10を接続する外部バス11と、ブリッジ4とクロック制御部2との間でデータをやり取りするクロック制御バス12を備えている。

【0016】このコンピュータシステムにおいては、CPU1は、ROM6や外部記憶装置10に記憶されているプログラムを入力部7や通信処理部9などから入力される信号に基づいて実行する。このとき、クロック制御部2は、ソフトウェアまたはハードウェアによってCPU1の負荷を監視して、その状況に応じて、CPU1の動作/停止信号を制御する。

【0017】図2は、本発明のクロック制御部2の構成を示すブロック図である。図2において、このクロック制御部2は、ブリッジ4 (図1参照) 経由でのCPU1 (図1) との命令やデータの受け渡し、I/F (Interface/Face) となるブリッジI/F (Inter/Face) 部20と、クロック制御部2全体を制御する制御部21と、CPU1 (図1) の動作/停止の制御を行うCPU制御部33と、動作タイミングを計るために基準クロックを計数するカウンタA22と、制御部21によって設定され

るCPU1 (図1) の間欠動作比のうち非動作の比率値を保持するレジスタA23と、レジスタA23の非動作の設定値とカウンタA22から出力されるカウント値とを比較するコンパレータA30と、動作タイミングを計るために基準クロックを計数するカウンタB24と、制御部21によって設定されるCPU1 (図1) の間欠動作比のうち動作の比率値を保持するレジスタB25と、レジスタB25の動作の設定値とカウンタB24から出力されるカウント値とを比較するコンパレータB31と、制御部21からの制御信号に基づいて間欠動作要求 (SLOWCLKREQ) 信号を出力するレジスタC26と、制御部21からの制御信号に基づいて動作停止要求 (STOPCLKREQ) 信号を出力するレジスタD27と、外部からの割込み信号及び入力部7 (図1) や出力部8 (図1)、通信処理部9 (図1)、外部記憶装置10 (図1) 等へのI/Oアクセスなどのイベント情報に応じたイベント信号 (EVENT#) をトリガ信号としてレジスタD27に出力するイベント監視部28と、制御部21の制御信号に基づいてイベント監視部28からイベント情報をリードし保持するレジスタE29と、コンパレータA30、コンパレータB31、レジスタC26及びレジスタD27から出力される信号に基づいてCPU1 (図1) の動作を停止させるためのクロック停止信号 (STOPCLK#) を生成してCPU制御部33などに出力するSTOPCLK#信号生成部32と、CPU制御部33からのGRANT信号の反転信号とSTOPCLK#信号生成部32からのSTOPCLK#信号の反転信号を論理積するANDゲート34と、ANDゲート34からの出力信号とSTOPCLK#信号生成部32からのSTOPCLK#信号を論理和して反転出力するNORゲート35とを備えている。

【0018】ここで、レジスタA23及びレジスタB25に保持される間欠動作の比は、CPU1 (図1) 稼働時の基準クロックの分周比やデューティ比に基づいて、ブリッジI/F (Inter/Face) 部20をとおして、ソフトウェア等でプログラマブルに決定することができる。

【0019】CPU制御部33は、STOPCLK#信号生成部32から出力されるSTOPCLK#信号に基づいて、CPUSTOPCLK#をCPU1 (図1) に出力する。また、CPU制御部33は、CPU1 (図1) から内部バス13を通してCPU1 (図1) の状態を受け取りGRANT信号をANDゲート34に出力するとともに、レディ信号 (RDY#) をCPU1 (図1) に出力する。

【0020】カウンタA22は、STOPCLK#信号生成部32からCPU制御部33に出力されるSTOPCLK#信号がアクティブ (ローレベル) であるとき、CPU制御部33から出力されるGRANT信号がアクティブ (ハイレベル) の場合 (つまりSTOPCLK#信号が「ロー」レベルでかつCPU1が停止し

ている間)にのみカウントを続行する。また、STOPCLK#信号生成部32及びCPU制御部33からの出力信号が上記以外の場合には、カウンタA22は、CLR端子が「ロー」レベルとなり、カウントをリセットする。一方、カウンタB24は、STOPCLK#信号生成部32からCPU制御部33へ出力されるSTOPCLK#信号がインアクティブ(「ハイ」レベル)の場合にカウントを続行し、STOPCLK#信号がアクティブ(「ロー」レベル)の場合にはカウントをリセットする。

【0021】コンパレータA30は、カウンタA22からの出力値であるカウント値とレジスタA23の設定値とを比較し、カウント値と設定値が一致すると、STOPCLK#信号生成部32のA端子に「ハイ」レベルを出力する。一方、コンパレータB31は、カウンタB24からの出力値であるカウント値とレジスタB25の設定値とを比較し、カウント値と設定値が一致すると、STOPCLK#信号生成部32のB端子に「ハイ」レベルを出力する。

【0022】イベント監視部28は、レジスタE29から出力されるEVENTCLR#信号がアクティブ(「ロー」レベル)になった時にリセットされる。また、レジスタD27は、イベント監視部28から出力されるEVENT#信号がアクティブ(「ロー」レベル)になるとリセットされる。

【0023】図3は、STOPCLK#信号生成部32の具体的な回路構成を示す図である。STOPCLK#信号生成部32(図2)は、コンパレータA30及びコンパレータB31(図2)からのパルス信号に基づいて信号を出力するトグル回路40と、トグル回路40からの出力信号及びレジスタC26(図2)から出力されるSLOWCLKREQ信号を入力して、論理積反転出力するNORゲート41と、ANDゲート41からの出力される信号及びレジスタD27(図2)から出力されるSTOPCLKREQ信号を入力して、論理積反転出力するNORゲート42とから構成される。

【0024】トグル回路40は、コンパレータA30(図2)から「ハイ」のバルス信号が出力されると出力を「ハイ」に固定し、コンパレータB31(図2)から「ハイ」のバルス信号が出力されると出力を「ロー」に固定する。

【0025】以下、図1～図3で示したコンピュータシステムの機能について説明する。

【0026】図4は、本発明のコンピュータシステムにおけるCPU1の動作状態と動作モードとの関係を示す概念図である。図4に示すとおり、CPU1の動作状態は、「ビジー状態」と「アイドル状態」に分類される。「ビジー状態」は、画像処理や文字認識、数値計算などのようにCPU1の負荷が大きい場合であり、「アイドル状態」は、キー入力待ち状態などCPU1の負荷が小

さい場合である。

【0027】制御部21は、ブリッジ部4から与えられるCPU1の動作状態を検出し(ステップ401)、CPU1の動作状態が変化したかを判断する(ステップ402)。CPU1の動作状態が変化していない場合は、処理を終了する。またCPU1の動作状態が変化していた場合には、現在のCPU1の動作状態を判断する(ステップ403)。CPU1の動作状態が「ビジー状態」の場合は、(A)ビジーモードの処理を実行する(ステップ404)。「アイドル状態」の場合は、「アイドル状態」を所定の閾値(CPUの稼働率など)に基づいて判断し(ステップ405)、(B)完全停止モード(ステップ406)または(C)間欠動作モード(ステップ407)の処理をそれぞれ実行する。

【0028】制御部21は、検出したCPU1の動作状態である稼働率などから、所定の閾値に基づいて、CPU1の動作モード及びアイドルの状態を判断する。例えば、基準クロックの分周比やデューティ比を稼働率として、予めシステム構成に基づいて決定した閾値と稼働率を比較することによって、CPU1の状態を判断するようにしてもよい。

【0029】以下に、各CPU動作モードにおける本発明のコンピュータシステムの処理の流れを詳細に説明する。

【0030】(A)ビジーモード

図5は、ビジーモードにおける本発明のコンピュータシステムの処理の流れを示すフローチャートである。図5において、制御部21は、レジスタC26に制御信号を出力し、レジスタC26から出力されるSLOWCLKREQ信号をインアクティブ(「ロー」レベル)にリセットする(ステップ501)。従って、CPU1は間欠動作や停止動作ではなく、通常の動作状態で稼働する。

【0031】(B)完全停止モード

図6は、完全停止モードにおける本発明のコンピュータシステムの処理の流れを示すフローチャートである。図6において、まず、制御部21がレジスタE29に制御信号を出力して、レジスタE29から出力されるEVENTCLR#信号をアクティブ(「ロー」レベル)に変化させ、イベント監視部28をリセットする(ステップ601)。

【0032】次に、制御部21がレジスタE29に制御信号を出力すると、レジスタE29はEVENTCLR#信号をインアクティブ(「ハイ」レベル)状態でイベント監視部28に出力する。このとき、イベント監視部28のリセット状態は解除される(ステップ602)。

【0033】次に、イベント監視部28でイベントを検出した場合(ステップ603)、イベント監視部28はEVENT#信号をアクティブ(「ロー」レベル)にセットして、レジスタD27をリセット状態にし(ステップ604)、完全停止モードには入らず処理を終了す

る。

【0034】一方、イベント監視部28でイベントを検出していない場合（ステップ603）、制御部21からの制御信号で、レジスタD27から出力されるSTOPCLKREQ信号がアクティブ（「ハイ」レベル）となる（ステップ605）。STOPCLK#信号生成部32は、NORゲート42でレジスタD27から出力されたSTOPCLKREQ信号の「ハイ」レベルを受け取ると、STOPCLK#がアクティブ（「ロー」レベル）となり、その信号をCPU制御部33に出力する（ステップ606）。CPU制御部33は、STOPCLK#信号生成部32から、アクティブ（「ロー」レベル）状態のSTOPCLK#を受け取ると、所定のCPU停止処理を行う（ステップ607）。これにより、CPU1は完全停止する（ステップ608）。

【0035】ここで、イベント監視部28が、予めイベント監視部28内部に登録しておいた監視すべき割り込み信号やI/Oアクセスなどを検出すると（ステップ609）、イベント監視部28は、EVENT#信号をアクティブ（「ロー」レベル）にする（ステップ610）。レジスタD27は、CLR端子に「ロー」レベルの信号が入力されるとリセットされ、レジスタD27から出力されるSTOPCLKREQ信号はインアクティブ（「ロー」レベル）にリセットされる（ステップ611）。一方、レジスタE29は、イベント情報を受け取ると、イベント情報を内部に保存後、イベント監視部28に対してEVENTCLR#信号をアクティブ（「ロー」レベル）にして、EVENT#信号とイベントステータスをリセットする（ステップ612）。また、レジスタE29は、イベント監視部28のイベントステータス信号がリセットされると、EVENTCLR#信号をインアクティブ（「ハイ」レベル）に戻す（ステップ613）。STOPCLK#信号生成部32は、NORゲート42でレジスタD27から出力されたSTOPCLKREQ信号（「ロー」レベル）を受け取ると、ANDゲート41からの「ロー」レベルの信号に応じてSTOPCLK#がインアクティブ（「ハイ」レベル）状態となってCPU制御部33に出力される（ステップ614）。CPU制御部33は、STOPCLK#信号生成部32から、インアクティブ（「ハイ」レベル）状態のSTOPCLK#を受け取ると、CPU動作開始処理を行う（ステップ615）。これにより、CPU1は動作を開始する（ステップ616）。

【0036】（C）間欠動作モード

図7は、間欠動作モードにおける本発明のコンピュータシステムの処理の流れを示すフローチャートである。図7において、まず、制御部21は、ブリッジ部4から与えられるCPU1の動作状態に基づいて間欠動作比率を設定し、レジスタA23に非動作の比率値、レジスタB25に動作の比率値を出力する（ステップ701）。レジ

スタA23及びレジスタB25の値に基づいてコンパレータA30及びコンパレータB31が動作し、それぞれの比較結果信号をSTOPCLK#信号生成部32のトグル回路40に出力する。ここで、比較結果信号は、レジスタの設定値とカウンタの値が一致した場合に「ハイ」レベルとなる。トグル回路40は、コンパレータA30からの信号で「ハイ」レベルの信号を出力し、コンパレータB31からの信号で「ロー」レベルの信号を出力する。

【0037】次に、制御部21は、レジスタC26に制御信号を出力し、レジスタC26から出力されるSLOWCLKREQ信号をアクティブ（「ハイ」レベル）に設定する（ステップ702）。STOPCLK#信号生成部32において、ANDゲート41は、レジスタC26から出力されたSLOWCLKREQ信号がアクティブ（「ハイ」レベル）の時、トグル回路40からの「ハイ」レベルの出力信号に応じて、「ハイ」レベルの信号を出力する。NORゲート42は、レジスタD27からのSTOPCLKREQ信号に応じてSTOPCLK#信号をCPU制御部33に出力する（ステップ703）。

【0038】CPU制御部33は、STOPCLK#信号生成部32からSTOPCLK#信号がアクティブ（「ロー」レベル）で出力されると（ステップ704）、所定のCPU停止処理（ステップ705）を行い、CPU1は動作を停止する（ステップ706）。

【0039】ここで、イベント監視部28において外部からの割り込み信号やI/Oアクセスなどを検出すると（ステップ707）、制御部21は、レジスタC26に制御信号を出力し、レジスタC26から出力されるSLOWCLKREQ信号をインアクティブ（「ロー」レベル）にリセットする（ステップ713）。SLOWCLKREQ信号がリセットされると、CPU制御部33には、インアクティブ（「ハイ」レベル）のSTOPCLK#信号が入力され、CPU制御部33は、所定のCPU動作開始処理を行う（ステップ714）。これにより、CPU1は動作を開始する（ステップ715）。また、制御部21は、イベント監視部28をリセット（ステップ716）した後、イベント監視部28のリセットを解除する（ステップ717）。これにより、間欠動作モードは終了する。

【0040】一方、イベント監視部28において外部からの割り込み信号やI/Oアクセスなどが検出されない状態で（ステップ707）、CPU制御部33に入力されているSTOPCLK#がインアクティブ（「ハイ」レベル）へと変化すると（ステップ708）、CPU制御部33は、所定のCPU動作開始処理を行う（ステップ709）。これにより、CPU1は動作を開始する（ステップ710）。ここで、イベント監視部28において外部からの割り込み信号やI/Oアクセスなどが検

出されず（ステップ711）、STOPCLK#信号がアクティブ（「ロー」レベル）になった場合（ステップ712）には、再びCPU制御部33によって、所定のCPU停止処理（ステップ705）を行い、CPU1は動作を停止する（ステップ706）。また、イベント監視部28において外部からの割り込み信号やI/Oアクセスなどを検出すると（ステップ711）、制御部21は、レジスタ26に制御信号を出し、レジスタ26から出力されるSLOWCLKREQ信号をインアクティブ（「ロー」レベル）にリセットする（ステップ713）。SLOWCLKREQ信号がリセットされると、CPU制御部33には、インアクティブ（「ハイ」レベル）のSTOPCLK#信号が入力され、CPU制御部33は、所定のCPU動作開始処理を行う（ステップ714）。これにより、CPU1は動作を開始する（ステップ715）。また、制御部21は、イベント監視部28をリセット（ステップ716）した後、イベント監視部28のリセットを解除する（ステップ717）。これにより、間欠動作モードは終了する。

【0041】また、ステップ707において、イベント監視部28で外部からの割り込み信号やI/Oアクセスなどが検出されない場合で、STOPCLK#がアクティブ（「ロー」レベル）の場合（ステップ708）には、引き続きイベント監視部28によってイベントの監視が行われる（ステップ707）。

【0042】同様に、ステップ711において、イベント監視部28で外部からの割り込み信号やI/Oアクセスなどが検出されない場合で、STOPCLK#がアクティブ（「ロー」レベル）の場合（ステップ715）には、引き続きイベント監視部28によってイベントの監視が行われる（ステップ711）。

【0043】図8は、所定のCPU停止処理の流れを示すフローチャートである。

【0044】CPU制御部33は、STOPCLK#信号生成部32から出力されるSTOPCLK#信号がアクティブ（「ロー」レベル）になると、CPUSTOPCLK#信号をアクティブ状態（「ロー」レベル）としてCPU1に出力する（ステップ801）。CPU1は、CPUSTOPCLK#信号がアクティブ（「ロー」レベル）となり、停止コマンドを受け付けると、CPUステータス（STOPCLK#受領状態）をCPU制御部33に出力する（ステップ802）。次に、CPU制御部33が、RDY#信号をアクティブ（「ロー」レベル）にしてCPU1に返す（ステップ803）。1CPUCLK後、CPU制御部33は、RDY#信号をインアクティブ（「ハイ」レベル）にリセットし、GRANT信号をアクティブ（「ハイ」レベル）にセットする（ステップ804）。これにより、CPU1は完全停止状態となる。

【0045】図9は、所定のCPU動作開始処理の流れ

を示すフローチャートである。

【0046】CPU制御部33は、STOPCLK#信号生成部32から出力されるSTOPCLK#信号がインアクティブ（「ハイ」レベル）になると、CPUSTOPCLK#信号をインアクティブ状態（「ハイ」レベル）としてCPU1に出力する（ステップ901）。CPU制御部33は、GRANT信号をインアクティブ（「ロー」レベル）にリセットする（ステップ902）。これにより、CPU1は動作を開始する。

【0047】このように、制御部21が、CPU1の負荷に応じてCPU1の動作状態を判断し、CPU動作停止信号（STOPCLK#）を複数のモードで変化させることによって、CPU1を完全停止状態から完全動作状態までの間でリニア（段階的）に制御することができる。

【0048】以上、本発明のコンピュータシステムについて説明したが、（B）完全停止モードや（C）間欠動作モードのとき、制御部21からイベント監視回路28に対して、所定のタイミングで割り込みを発生し、（B）完全停止モードや（C）間欠動作モードから（A）ビジーモードに回帰させるようにしてもよい。

【0049】また、制御部21に温度センサーやバッテリーセンサー等を装着し、CPU1の温度やバッテリー残量を監視し、これらのセンサーからの信号がある閾値を超えたら強制的に（C）間欠動作モードや（B）完全停止モードに移行させ、信号がある閾値より下がったら、（A）ビジーモードに回帰させるようにしてもよい。

【0050】

【発明の効果】以上のように、本発明のコンピュータシステムによれば、当該コンピュータシステムのパフォーマンスを低下させることなく、その消費電流を低減することができるようになった。

【図面の簡単な説明】

【図1】本発明のコンピュータシステムの構成の一例を示す図である。

【図2】本発明のクロック制御部の構成を示すブロック図である。

【図3】STOPCLK#信号生成部の回路構成を示す図である。

【図4】本発明のコンピュータシステムにおけるCPUの動作状態と動作モードとの関係を示す概念図である。

【図5】ビジーモードの処理の流れを示すフローチャートである。

【図6】完全停止モードの処理の流れを示すフローチャートである。

【図7】間欠モードの処理の流れを示すフローチャートである。

【図8】CPU停止処理の流れを示すフローチャートである。

【図9】CPU動作開始処理の流れを示すフローチャート

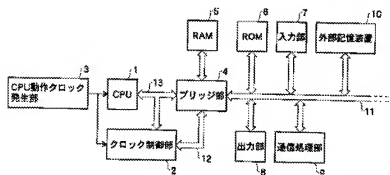
トである。

【符号の説明】

- 1 CPU
- 2 クロック制御部
- 3 CPU動作クロック発生部
- 4ブリッジ部
- 5 RAM
- 6 ROM
- 7 入力部
- 8 出力部
- 9 通信処理部
- 10 外部記憶装置
- 11 外部バス
- 12 クロック制御バス
- 13 内部バス
- 20 ブリッジ1ノF部

- 21 制御部
- 22 カウンタA
- 23 レジスタA
- 24 カウンタB
- 25 レジスタB
- 26 レジスタC
- 27 レジスタD
- 28 イベント監視部
- 29 レジスタE
- 30 コンパレータA
- 31 コンパレータB
- 32 STOPCLK#信号生成部
- 33 CPU制御部
- 34、41 ANDゲート
- 35、42 NORゲート
- 40 トグル回路

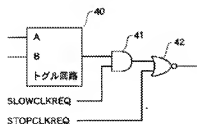
【図1】



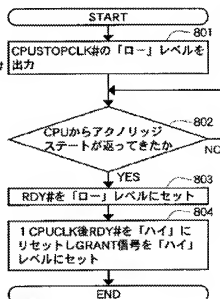
【図5】



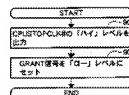
【図3】



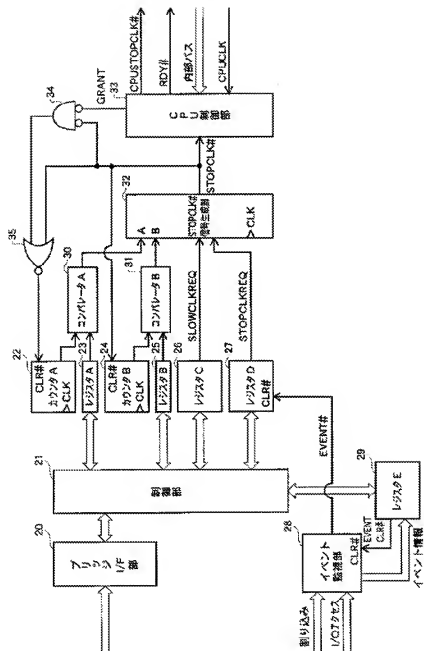
【図8】



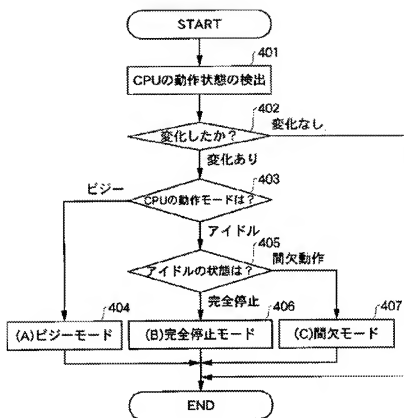
【図9】



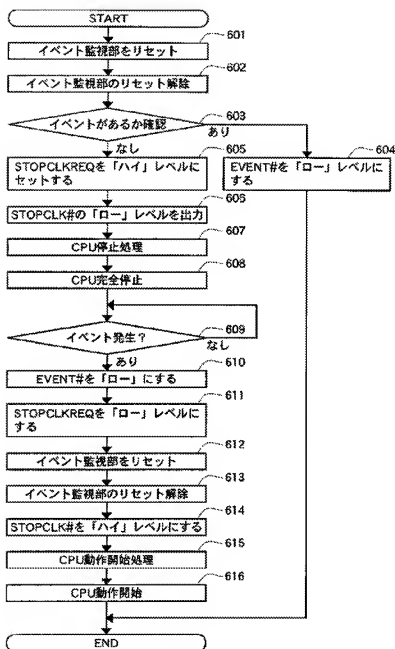
【図 2】



【図4】



【図6】



【図7】

